```
#!/usr/local/bin/perl
# COPYRIGHT  (c) 1998 Bell Communications Research Inc.,
#   All Rights Reserved.
#
# PROPRIETARY - BELLCORE AND AUTHORIZED CLIENTS ONLY.
#
# This document contains proprietary information that shall
# be distributed or routed only within Bell Communications
# Research (Bellcore), and its authorized clients, except
# with written permission of Bellcore.
#
# $Id: getCovar.pl,v 1.2 1999/05/25 15:33:47 rmartija Exp rmartija $
#

undef;
require 'getopts.pl';
require '/u/rmartija/netsizer/scripts/math.pl';

$USAGE = "Usage: " . $0 . " [-D] -d domain  file\n\n" .
         "Options:\n" .
         "    -D           debug mode\n" .
         "    -d domain    domain type (1=US, 2=Non-US)\n" .
         "    file         name of input file. The default is STDIN.\n\n" .
         "Example:\n" .
         "   $0 ../data/test.out\n" .
         "   $0 -d 1 ../data/test.out\n" .
         "   $0 -D ../data/test.out\n" .
         "   $0 -D -d2 ../data/test.out\n\n" ;


#################################################################
####################### main program ###########################
#################################################################

$x = &Getopts( 'd:D' );
die "$USAGE\n" unless ($x ne '');

die "$USAGE\n" unless $opt_d && $opt_d >= 1 && $opt_d <= 2;

if( $opt_d == 1 ) {
    $domain = 'US';
}
else {
    $domain = 'NONUS';
}

$oldLoc = '';
$rows = 0;
$cols = 0;

die "$USAGE\n" if( $#ARGV > 0 );

if( $#ARGV < 0 || $ARGV[0] eq '-' ) {
    $INPUT = STDIN;
}
else {
        die "ERROR: cannot open $ARGV[0]\n" unless -r $ARGV[0];
        open( INPUT, "< $ARGV[0]" );
        $INPUT = INPUT;
    }

    while( <$INPUT> ) {
        chop;
        next unless length($_) > 0;
        @tokens = split( '\t', $_ );
        $locale = $tokens[0];

        if( $locale ne $oldLoc ) {
            if( $oldLoc ne '' ) {
                %m = &getMeans( $rows-1, $cols, *matrix );
                print "$domain: $oldLoc\n";
                print "MEAN: " ;
                for( $i = 1; $i <= $cols; $i++ ) {
                    printf "%.2f", $m{$i} ;
                    print " " if( $i < $cols );
                    print "\n" if( $i == $cols );
                }

                if( $opt_D ) {
                    print "ORIGINAL MATRIX:\n" ;
                    for( $i = 1; $i <= $cols; $i++ ) {
                        for( $j = 1; $j <= $cols; $j++ ) {
                            printf "%12.2f", $matrix{$j + (($i - 1) * $cols)} ;
                            print " " if( $j < $cols );
                            print "\n" if( $j == $cols );
                        }
                    }
                    print "\n" ;
                }

                %S = &getCovarianceMatrix(  $rows-1, $cols, *matrix, *m );
                if( $opt_D ) {
                    print "COVARIANCE MATRIX:\n" ;
                    for( $i = 1; $i <= $cols; $i++ ) {
                        for( $j = 1; $j <= $cols; $j++ ) {
                            printf "%12.2f", $S{$j + (($i - 1) * $cols)} ;
                            print " " if( $j < $cols );
                            print "\n" if( $j == $cols );
                        }
                    }
                    print "\n" ;
                }

                %I = &getInverseMatrix( $cols, *S );
                print "INVERSE OF COVARIANCE MATRIX:\n" ;
                for( $i = 1; $i <= $cols; $i++ ) {
                    for( $j = 1; $j <= $cols; $j++ ) {
                        printf "%12.2f", $I{$j + (($i - 1) * $cols)} ;
                        print " " if( $j < $cols );
                        print "\n" if( $j == $cols );
                    }
                }
                print "\n";
        }

        $oldLoc = $locale;
        $rows = 1;
        $cols = @tokens - 1;
    }

    for( $j = 1; $j <= $cols; $j++ ) {
        $matrix{$j + (($rows - 1) * $cols)} = $tokens[$j] * 1.0;
    }

    $rows++;
}

close( $INPUT ) unless $#ARGV < 0 || $ARGV[0] eq '-';

%m = &getMeans( $rows-1, $cols, *matrix );
print "$domain: $oldLoc\n";
print "MEAN: " ;
for( $i = 1; $i <= $cols; $i++ ) {
    printf "%.2f", $m{$i} ;
    print " " if( $i < $cols );
    print "\n" if( $i == $cols );
}

if( $opt_D ) {
    print "ORIGINAL MATRIX:\n" ;
    for( $i = 1; $i <= $cols; $i++ ) {
        for( $j = 1; $j <= $cols; $j++ ) {
            printf "%12.2f", $matrix{$j + (($i - 1) * $cols)} ;
            print " " if( $j < $cols );
            print "\n" if( $j == $cols );
        }
    }
    print "\n" ;
}

%S = &getCovarianceMatrix(  $rows-1, $cols, *matrix, *m );
if( $opt_D ) {
    print "COVARIANCE:\n" ;
    for( $i = 1; $i <= $cols; $i++ ) {
        for( $j = 1; $j <= $cols; $j++ ) {
            printf "%12.2f", $S{$j + (($i - 1) * $cols)} ;
            print " " if( $j < $cols );
            print "\n" if( $j == $cols );
        }
    }
    print "\n" ;
}

%I = &getInverseMatrix( $cols, *S );
print "INVERSE OF COVARIANCE MATRIX:\n" ;
for( $i = 1; $i <= $cols; $i++ ) {
    for( $j = 1; $j <= $cols; $j++ ) {
        printf "%12.2f", $I{$j + (($i - 1) * $cols)} ;
        print " " if( $j < $cols );
        print "\n" if( $j == $cols );
        }
    }
```